

論理の直並列分解による回路の最適設計法 に関する研究

松崎 英樹

明治大学情報科学センター／教育研究システム課

あ ら ま し

論理回路を容易に実現できる PLA(Programmable Logic Array)は、近年幅広く用いられている。しかし PLA には実際の論理の実現に寄与しない未使用領域が多く存在するため、回路面積が大きくなるという欠点がある。また、複数の PLA を 1 つのチップに集積した CPLD (Complex Programmable Logic Device)では、CPLD 内の個々の PLA には外部入出力数などに制限があるため、大規模な論理をそのままでは実現できない場合がある。これらの問題を解決する 1 つの方策として、元の PLA を複数の小さな PLA に分解する手法がよく知られている。

本稿では、直-並列分解および並-直列分解の手法が PLA の入出力数や面積の削減に有効であることを示す。また、直列分解および並列分解が PLA のランダムテストのテストビリティに及ぼす影響について評価した結果を示し、直列分解がランダムテストにとって有効であることを示す。

A Design for the Logic Circuit Optimization by Decomposition

Hideki Matsuzaki

Meiji University, Information Science Center/Education and Research System Section

Abstract

It is easy to realize logic circuits with PLA(Programmable Logic Array) , and recently PLA is used in many ways. But PLA has much unused area which does not contribute to logic realization, so it makes chip area to increase. And in some cases, CPLD(Complex Programmable Logic Device) into which some PLAs are integrated is not able to be used to realize a large logic, because each PLA block in CPLD has limit of the number of primary inputs/outputs. It is well known as a method to solve these problems to decompose a large PLA into a set of interconnected smaller PLA blocks.

This paper describes that PLA serial-parallel/parallel-serial decomposition is useful to reduce PLA inputs/outputs and its area. In addition, it discusses how the random pattern testability is improved with the serial/parallel decomposition, based on the experimented results, and shows the serial decomposition is available.

キーワード： テスタビリティ, PLA, CPLD, 論理関数分解

1.はじめに

PLA は規則的な構造をもっているため設計が容易であり、大規模な論理回路を短期間で実現する上で有効なデバイスの1つである。しかし PLA には、実際の論理の実現に寄与しない未使用領域が多数存在するため回路面積が大きくなるという欠点がある。また、複数の PLA を1つのチップに集積した CPLD (Complex Programmable Logic Device) が近年幅広く利用されているが、CPLD 内の個々の PLA には外部入出力数などに制限があるため、大規模な論理をそのままでは実現できない場合がある。これらの問題を解決する1つの方策として、元の PLA を複数の小さな PLA に分解し、それらを相互に接続することで全体の回路を実現する手法がよく知られている[1]-[8]。

PLA の分解には、直列分解と並列分解がある。直列分解の手法としては、論理関数分解を用いた手法が提案されている[1]-[5]。Weak Division を基本とした手法[3]は、Implicit Don't Care を用いた単純化[4][5]ができ、比較的効率良く処理が可能である。並列分解の手法としては、畳み込みを活用した手法[6]などが提案されている。また、これらの手法を組み合わせた分解のためのツールとして PLASTIC システム[8]が提案されている。PLASTIC システムでは、多段構成を用いて PLA の交点数を削減しているが、多段構成を用いた場合、信号遅延が増大するという問題が生じる。

ここでは、従来の分解手法を組み合わせ、ベンチマーク PLA を直-並列分解および並-直列分解したときの、入力数、出力数、積項線数および交点数削減率について評価した結果を示し、直-並列分解および並-直列分解が PLA の入出力数や面積の削減に有効であることを示す。また、PLA は一般にファンイン数が多いことからランダムパターンに対するテストバリエーションが悪いといわれているが、分解によりファンイン数を小さく抑えることができれば、テストバリエーションを改善できる可能性がある[10],[11]。そこで、直列分解および並列分解が PLA のランダムテストのテストバリエーションに及ぼす影響について評価した結果を示し、直列分解がランダムテストにとって有効であることを示す。

2. PLA とは

PLA は AND-OR の2段論理を実現するものであり、一般的に、デコーダ、AND アレイ、OR アレイより構成される。図 1(a)は次の関数 $f1$ および $f2$ を実現した PLA である。

$$f1 = \bar{a}\bar{b}f\bar{g} + \bar{a}c\bar{f}\bar{g} + \bar{a}c\bar{e}\bar{f} + \bar{a}c\bar{e}f + d\bar{e}f\bar{g} + \bar{c}\bar{e}f\bar{g}$$

$$f2 = \bar{a}\bar{b}e\bar{h} + \bar{a}c\bar{e}h + d\bar{e}h\bar{i} + \bar{c}\bar{e}h\bar{i} + d\bar{e}h\bar{i} + \bar{c}\bar{e}h\bar{i}$$

$a \sim i$ の各入力は、デコード線 (AND アレイ部の水平線) により、その肯定値および否定値が AND アレイに供給される。同図の 1~12 の番号が付いた垂直線は積項線と呼ばれ、AND デバイスで接続されているデコード線の積すなわち上式中の積項を表す。例えば積項線 1 は積項 $\bar{a}\bar{b}f\bar{g}$ を表している。出力線 (OR アレイ部の水平線) では、OR デバイスで接続された積項線の和が得られ、 $f1$ および $f2$ の出力となる。

PLA のチップ上に占める面積は、PLA の規模が大きくなるに従って、デコード部分に比べて AND アレイおよび OR アレイ部分の割合が高くなる。アレイ部分の面積は、デコード線と積項線との交点および積項線と出力線との交点の数に比例することから、PLA の面積を論じるときは交点数を指標とすることが多い。本稿でもこれを用いることにする。

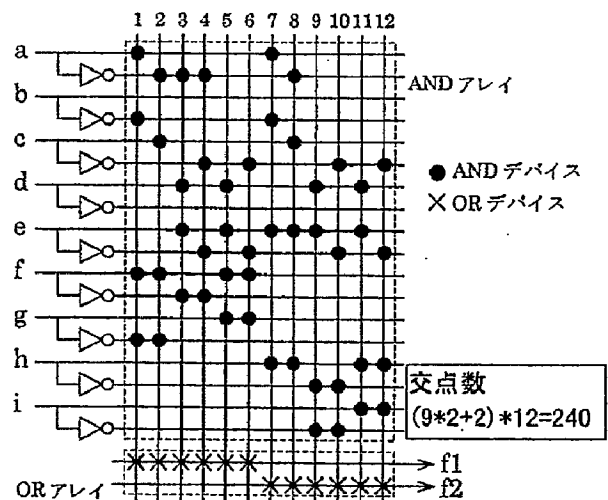


図 1(a) 分解前の PLA

3. PLA の直列分解と並列分解

PLA の直列分解とは、1つの PLA を2つの PLA に分解し、一方の PLA の出力が他方の PLA の入力の一部もしくは全部となるような構成に置き換えることをいう。また、並列分解とは、外部入力を同時に複数の PLA に加えるような構成に置き換えることをいう。

従来、様々な分解手法が提案されているが、以下では Weak division を基本とする直列分解手法[3]および畳み込みを活用した並列分解手法[6]について、例を用いて述べる。

Weak division を基本とする直列分解手法では、論理関数を代数式で表現し、カーネルと呼ばれる共通項を独立の前段の PLA として分解する。

[例 1] 図 1(a)の関数 $f1$ および $f2$ が与えられているとする。これらは次のように変形できる。

$$f1 = (ab + ac)fg + (de + ce)(af + fg)$$

$$f2 = (ab + ac)eh + (de + ce)(hi + hi)$$

いま、カーネルとして、

$$v1 = ab + ac$$

$$v2 = de + ce$$

を選んだとすると、

$$f1' = v1 \cdot fg + v2 \cdot (af + fg)$$

$$f2' = v1 \cdot eh + v2 \cdot (hi + hi)$$

と表すことができる。関数 $v1, v2$ を前段、関数 $f1', f2'$ を後段とすることにより、図 1(b)の PLA が得られる。この PLA は図 1(a)の PLA と機能的に等価であるが、交点数が 240 から 156 に削減されている。 ■

畳み込みを活用した並列分解手法では、共有する入力および出力をもたない積項線を元に2つの独立した PLA に分解する。

[例 2] 図 1(a)の PLA では、積項線 1,2 と 9,11 が入力も出力も共有していない。そこで、これらの積項線を別々の PLA に配置し、2つの PLA がそれぞれ $f1, f2$ を実現するように、その他の積項線を適切に振り分ける。このようにして得られたのが、図 2(b)に示す PLA である。これは図 1(a)の PLA と機能的に等価であるが、交点数が 240 から 180 に削減されている。 ■

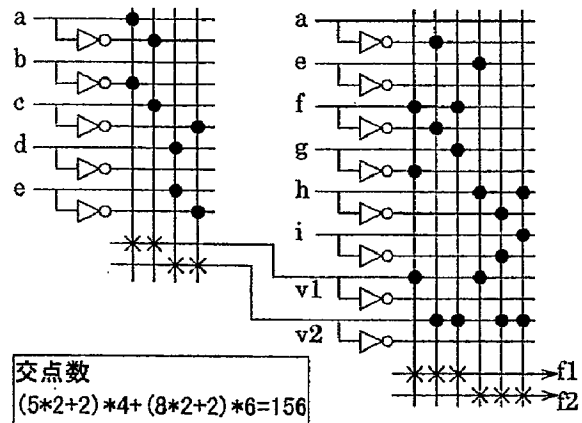


図 1(b) 直列分解後

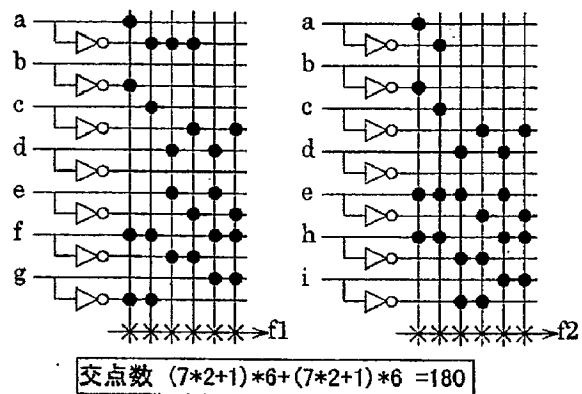


図 1(c) 並列分解後

4. 直-並列分解と並-直列分解

直-並列分解とは、元の PLA をまず直列に分解し、分解して得られた2つの PLA をそれぞれ並列に分解することをいう。

[例 3] 図 1(a)の PLA を直列分解して得られた同図(b)の PLA をさらに並列に分解すると、図 2 の PLA が得られる。交点数は図 1(b)に示す直列分解のみのときが 156 であるのに対し、さらに 94 に削減されている。 ■

並-直列分解では、元の PLA を最初に並列に分解し、その後で個々の PLA を直列に分解する。

[例 4] 図 1(a)の PLA を並列分解して得られた同図(c)の PLA をさらに直列に分解すると、図 3 の PLA が得られる。交点数は図 1(c)に示す並列分解のみのときが 180 であるのに対し、さらに 162 に削減されている。 ■

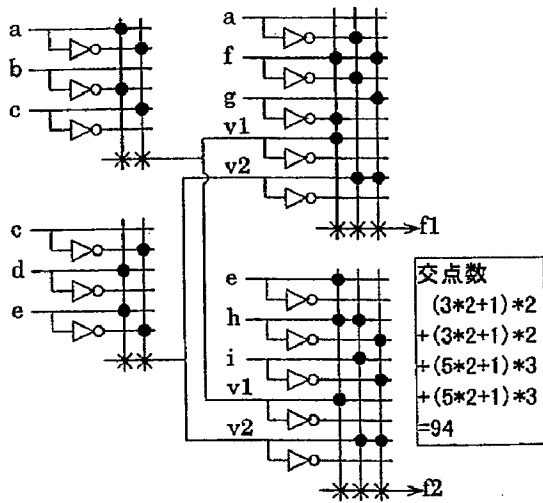


図2 直-並列分解後

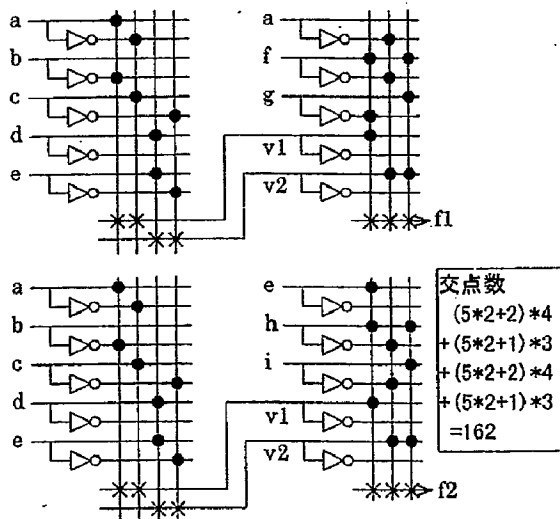


図3 並-直列分解後

5. Implicit Don't Care を用いた簡単化

Weak Division を基本とする直列分解では、一般に前段と後段で共通の外部入力が存在する。そのため、後段に加えられることのない入力の組み合わせが存在することがある。このような入力の組み合わせを、SDC (Satisfiability Don't Care) と呼ぶ。この SDC を利用した後段の PLA の簡単化について以下に例を用いて述べる。

[例 5] 次の関数 $f1$ および $f2$ が与えられているとする。

$$f1 = abcd + \bar{a}bcd + ab\bar{d} + \bar{a}b\bar{d} + \bar{a}cd$$

$$f2 = abcd + \bar{a}bcd + ab\bar{c} + \bar{a}b\bar{c} + \bar{a}cd$$

関数 $f1, f2$ を PLA で実現したものを図 4(a) に示す。ここでカーネルとして、

$$y = ab + \bar{a}\bar{b} \quad (\text{前段})$$

を選んだとすると、

$$f1' = ycd + \bar{a}bd + \bar{a}b\bar{d} + \bar{a}cd \quad (\text{後段})$$

$$f2' = ycd + \bar{a}b\bar{c} + \bar{a}b\bar{c} + \bar{a}cd$$

となる。この直列分解後の PLA を図 4(b) に示す。このとき後段に決して生じない入力の組み合わせは、

$$SDC = (ab + \bar{a}\bar{b}) \oplus y$$

と表現される。 $f1$ のカルノー図に SDC を表現したものを図 5 に示す。これにより $f1$ は、

$$f1 = ycd + \bar{y}d + \bar{a}cd$$

と簡単化できる。 $f2$ についても同様に簡単化でき、次の式が得られる。

$$f2 = ycd + \bar{y}c + \bar{a}cd$$

このようにして得られた簡単化後の PLA を図 4(c) に示す。元の PLA において 80 であった交点数は、60 に削減されている。 ■

6. 評価結果

ベンチマーク PLA に対して分解を施す前後の、入力数、出力数、積項数、交点数および交点数削減率を評価した結果を表 1 に示す。今回の実験では、直列分解後は必ず SDC による簡単化を行なうこととした。また各ベンチマーク PLA は、分解前および分解後に必ず ESPRESSO-II[9]を用いて簡単化を行なうこととした。

多くの PLA では、直列分解、並列分解のどちらか一方のみを行なったときよりも、直-並列分解もしくは並-直列分解を行なった方が、交点数の削減率が高くなっている。このことから、大規模な PLA に単純な 2 段の直-並列分解もしくは並-直列分解を行なうことによって、効果的に面積縮小できるものと思われる。入力数、出力数、積項線数についてもほとんどの場合、直-並列分解もしくは並-直列分解を行った場合のほうが少なくなっている。

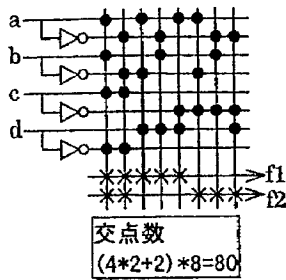


図 4(a) 分解前

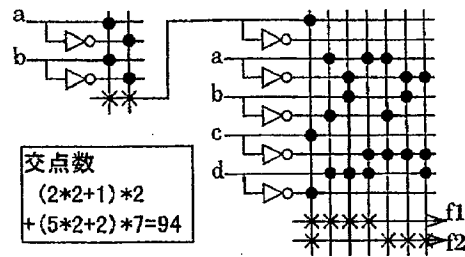


図 4(b) 直列分解後

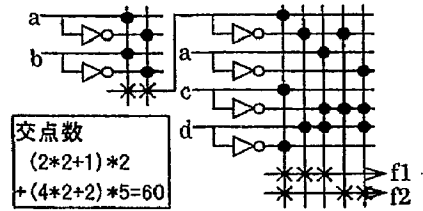


図 4(c) SDC による簡単化後

a	0	0	1	1
b	0	1	1	0
c	0	*	*	*
d	0	*	1	1
e	0	*	1	1
f	0	*	*	*

 $y=0$

a	0	0	1	1
b	0	1	1	0
c	0	*	*	*
d	0	1	1	1
e	0	1	1	1
f	0	1	1	1

 $y=1$

*: SDC

図 5 関数 $f1$ の SDC

表 1 評価結果

PLA	分解方法	入力線				出力線				積項線				交点数	削減率
p1	分解前	45				40				117				15210	
	直列	17		46		1		40		2		117		15514	-2.00%
	直-並列	17	0	40	39	1	0	28	12	2	0	75	85	15820	-4.01%
	並列	45		28		34		6		109		18		14632	3.80%
	並-直列	12	46	21	27	1	34	1	6	2	109	3	17	14933	1.82%
p4	分解前	32				11				124				9300	
	直列	19		35		11		11		24		47		4983	46.42%
	直-並列	10	12	31	32	6	5	6	5	14	11	27	25	4244	54.37%
	並列	30		31		6		5		41		88		8602	7.51%
	並-直列	9	29	13	33	4	6	5	5	8	29	15	36	5053	45.67%
p7	分解前	48				56				175				26600	
	直列	10		50		2		56		9		143		22506	15.39%
	直-並列	5	5	33	18	1	1	22	34	5	4	73	71	11493	56.79%
	並列	20		29		21		35		107		85		14432	45.74%
	並-直列	12	23	3	30	6	21	1	35	23	31	3	78	10198	61.66%
p9	分解前	62				10				279				37386	
	直列	16		62		7		10		31		153		21711	41.93%
	直-並列	12	11	38	39	3	4	7	3	14	17	54	61	10243	72.60%
	並列	40		35		4		6		141		138		22332	40.27%
	並-直列	16	41	11	29	8	4	4	6	28	34	13	24	5918	84.17%
p420	分解前	17				29				222				13986	
	直列	9		18		1		29		3		204		13317	4.78%
	直-並列	9	0	18	17	1	0	8	21	3	0	92	136	11585	17.17%
	並列	17		16		12		17		114		139		12055	13.81%
	並-直列	4	18	6	17	1	12	1	17	2	107	4	133	11989	14.28%
[3] Ex.	分解前	47				17				163				18093	
	直列	13		49		8		17		22		41		5463	69.81%
	直-並列	12	9	26	27	4	4	11	6	14	8	22	21	3214	82.24%
	並列	29		25		8		9		100		63		10317	42.98%
	並-直列	13	31	7	20	7	8	2	9	21	25	7	18	3437	81.00%

※交点数 = (入力数 * 2 + 出力数) * 積項数

※削減率 = (1 - 分解後における総交点数 / 分解前の総交点数) * 100%

7. 分解がテストビリティに及ぼす影響

PLA は、一般にファンイン数が多いことから、ランダムパターンに対するテストビリティが悪いといわれている。しかし、分解によりファンイン数を小さく抑えることができれば、テストビリティを改善できる可能性がある[10][11]。

ベンチマーク PLA について、分解前後のテストビリティについて評価した結果を表 2 に示す。表 2 では、元の PLA および並列分解後の PLA に対して ESPRESSO-II[9]による簡単化を行ない、直列分解後の PLA には SDC による Don't Care を考慮して ESPRESSO-II による簡単化を行なった。図 6 は、表 2 の vg2 について、テストパターン数に対する故障検出率を表したものである。

表 2 および図 6 より次のことがいえる。

- ・多くの場合、分解により冗長故障の数が減少する。
- ・直列分解により、元の PLA より少ないテスト

パターン数(数分の 1～数百分の 1)で同等の故障検出率が得られる。

- ・並列分解によるテストビリティの改善効果はあまりない。

図 7(a)および(b)は、vg2 の直列分解後および並列分解後における、積項線 1 本あたりのファンイン数に対する積項線数の分布を示したものである。同図(a)のように、元の PLA ではファンイン数の最大値が 18 であったが、直列分解後は 15 に減少している。この減少がテストビリティの向上につながったと考えられる。一方同図(b)に示されるように、並列分解では元の PLA 中の積項線をほぼそのまま 2 つに振り分けた形となっている。そのため、ファンイン数 18 および 17 の積項線がそのまま分解後の PLA に残存し、ファンインが減少していない。このことが並列分解によつてはテストビリティが改善しない理由と考えられる。

表 2 テスタビリティの評価

PLA		入力	出力	積項	冗長故障	検査対象故障	テストパターン数			
							故障検出率[%]			
							80.0	90.0	95.0	99.0～
ryy6	分解前	16	1	112	880	2,816	30,000	40,000	50,000	60,000
	直列 前段	8	3	8	16	136				
	分解 後段	11	1	7	20	141	300	300	500	600
	並列 1	-	-	-	-	-				
	分解 2	-	-	-	-	-	-	-	-	-
ts10	分解前	22	16	128	0	7,680	300	500	600	1,000
	直列 前段	20	16	64	0	3,584				
	分解 後段	18	16	32	0	1,664	80	200	300	400
	並列 1	21	8	64	0	3,200				
	分解 2	21	8	64	0	3,200	300	500	600	1,000
vg2	分解前	25	8	110	314	6,066	4,000	20,000	200,000	800,000
	直列 前段	11	2	10	4	236				
	分解 後段	16	8	22	54	826	300	2,000	9,000	40,000
	並列 1	25	2	50	110	2,490				
	分解 2	18	6	60	134	2,386	5,000	20,000	200,000	800,000
bc0	分解前	21	11	178	435	8,999	3,000	10,000	40,000	200,000
	直列 前段	5	1	5	20	35				
	分解 後段	22	11	173	526	8,989	3,000	7,000	30,000	200,000
	並列 1	18	2	59	117	2,125				
	分解 2	21	9	166	429	8,037	3,000	9,000	30,000	200,000

※ryy6については、出力線が1本しか存在しないため、並列分解不可能。

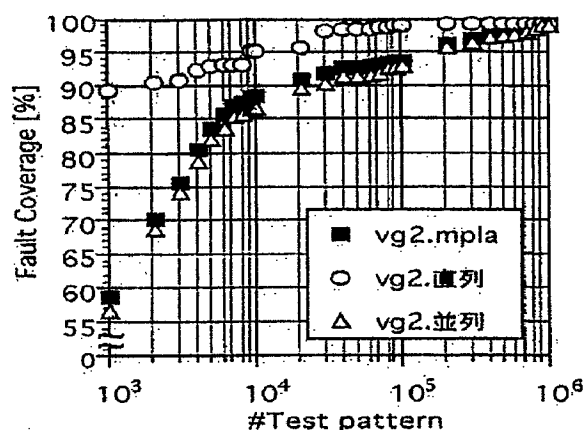
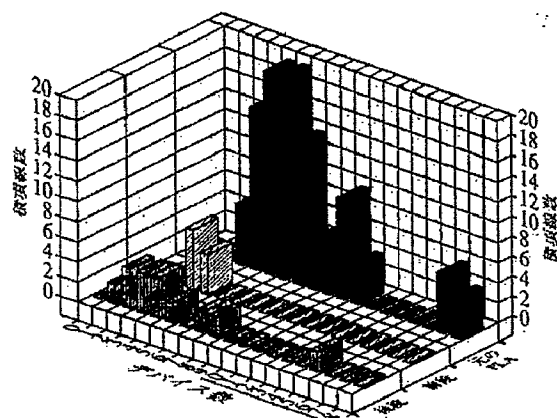
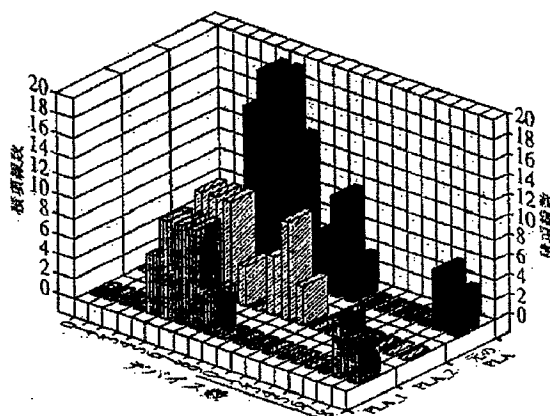


図6 テストパターン数に対する故障検出率



(a)直列分解



(b)並列分解

図7 分解後のPLAのファンイン数分布

8.むすび

従来の分解手法を組み合わせ、PLAを直-並列分解および並-直列分解したときの、入力数、出力数、積項線数および交点数削減率について評価した結果を示した。評価の結果、直-並列分解および並-直列分解が、PLAの入出力数や面積の削減に有効であることが明らかとなった。また、直列分解および並列分解がPLAのランダムパターンに対するテストビリティに及ぼす影響について評価した結果を示し、直列分解によってファンインの最大値を減少させることにより、ランダムパターンテストビリティの改

善が可能であることを示した。

今後は、直列分解後のPLAの簡単化にODC(Observability Don't Care)も考慮した場合について評価が必要である。また、大規模なPLAを分解する際、面積や遅延とともに、テストビリティを向上させることも考慮した分解手法を検討していく予定である。

謝辞 本研究を進めるにあたりご指導頂きました明治大学理工学部情報科学科の山田輝彦教授、井口幸弘講師、山崎浩二講師に深謝致します。

参考文献

- [1] 笹尾, 東田, "PLA の直列分解について", 信学技報, VLD 87-84, 1987.
- [2] S.Devadas, A.R.Wang, A.R.Newton and A.Sangiovanni-Vincentelli, "Boolean Decomposition of Programmable Logic Arrays", CICC'88, 2.5.1-5, 1988.
- [3] R.K.Brayton and C.T.McMullen, "The Decomposition and Factorization of Boolean Expressions", ISCAS'82, pp.49-54, May 1982.
- [4] 笹尾, 論理設計—スイッチング回路理論—, 10 章, 近代科学社, 1995.
- [5] K.A.Bartlett, R.K.Brayton, G.D.Hachtel, R.M.Jacoby, C.R.Marrison, A.Sangiovanni-Vincentelli and A.R.Wang, "Multi-Level Logic Minimization Using Implicit Don't Cares", IEEE Trans., CAD-7, 6, pp.723-740, June 1988.
- [6] 井口, 松島, 松崎, "畳み込みを活用した PLA の並列分割", 信学技報, VLD95-91, 1995.
- [7] 笹尾, 東田, "PLA の並列分解について", 信学技報, VLD 88-85, 1988.
- [8] S.Malik and R.H.Katz, "Combining Multi-Level Decomposition and Topological Partitioning for PLAs", Proc. of ICCAD'87, pp.112-115, 1987.
- [9] R.K.Brayton, G.D.Hachtel, C.T.McMullen and A.Sangiovanni-Vincentelli, "ESPRESSO-II: A New Logic Minimizer for Programmable Logic Arrays", Proc. of CICC'84, pp.370-376, 1984.
- [10] H.D.Schnurmann, E.Lindbloom and R.G.Carpenter, "The Weighted Random Test-Pattern Generator", IEEE Trans. Comput., Vol.C-24, pp.695-700, 1975.
- [11] E.B.Eichelberger and E.Lindbloom, "Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self-Test", IBM J.R.&D, Vol.27, pp.265-272, 1983.